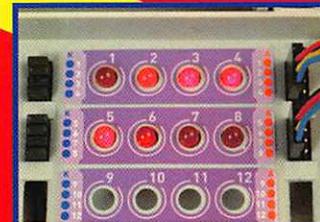
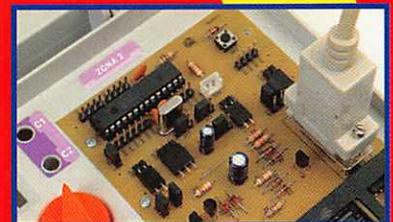
...e costruisci il tuo **LABORATORIO DIGITALE**

6,90 €

47

HARDWARE

DIGITALE DI BASE

MICROCONTROLLER

DIGITALE AVANZATO

TOTALMENTE PROGRAMMABILE!!!

9 771824 363008 41047

Peruzzo & C.

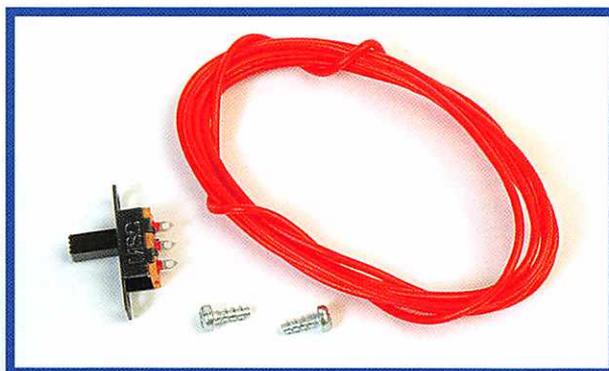
Direttore responsabile:
ALBERTO PERUZZO
Direttore Grandi Opere:
GIORGIO VERCELLINI
Consulenza tecnica e traduzioni:
CONSULCOMP S.n.c.
Pianificazione tecnica
LEONARDO PITTON

Direzione, Redazione, Amministrazione: viale Ercole Marelli 165, Tel. 02/242021, 20099 Sesto San Giovanni (Mi). Pubblicazione settimanale. Registrazione del Tribunale di Monza n. 1738 del 26/05/2004. Spedizione in abbonamento postale gr. II/70; autorizzazione delle Poste di Milano n. 163464 del 13/2/1963. Stampa: Grafiche Porpora s.r.l., Cernusco S/N (MI). Distribuzione SO.DI.P. S.p.A., Cinisello Balsamo (MI).

"ELETTRONICA DIGITALE" si compone di 70 fascicoli settimanali da suddividere in 2 raccoglitori.

RICHIESTA DI NUMERI ARRETRATI. Per ulteriori informazioni, telefonare dal lunedì al venerdì ore 9.30-12.30 all'ufficio arretrati tel. 02/242021. Se vi mancano dei fascicoli o dei raccoglitori per completare l'opera, e non li trovate presso il vostro edicolante, potrete riceverli a domicilio rivolgendovi direttamente alla casa editrice. Basterà compilare e spedire un bollettino di conto corrente postale a PERUZZO & C. s.r.l., Ufficio Arretrati, viale Marelli 165, 20099 Sesto San Giovanni (MI). Il nostro numero di c/c postale è 42980201. L'importo da versare sarà pari al prezzo dei fascicoli o dei raccoglitori richiesti, più le spese di spedizione € 3,10 per pacco. Qualora il numero dei fascicoli o dei raccoglitori sia tale da superare il prezzo globale di € 25,82 e non superiore a € 51,65, l'invio avverrà per pacco assicurato e le spese di spedizione ammonteranno a € 6,20. La spesa sarà di € 9,81 da € 51,65 a € 103,29; di € 12,39 da € 103,29 a € 154,94; di € 14,98 da € 154,94 a € 206,58; di € 16,53 da € 206,58 in su. Attenzione: ai fascicoli arretrati, trascorse dodici settimane dalla loro distribuzione in edicola, viene applicato un sovrapprezzo di € 0,52, che andrà pertanto aggiunto all'importo da pagare. Non vengono effettuate spedizioni contrassegno. Gli arretrati di fascicoli e raccoglitori saranno disponibili per un anno dal completamento dell'opera. IMPORTANTE: è assolutamente necessario specificare sul bollettino di c/c postale, nello spazio riservato alla causale del versamento, il titolo dell'opera nonché il numero dei fascicoli e dei raccoglitori che volete ricevere.

IN REGALO in questo fascicolo

1 Commutatore
1 Spezzone di filo flessibile rosso
2 Viti

IN REGALO nel prossimo fascicolo

1 CD-ROM

COME RACCOGLIERE E SUDDIVIDERE L'OPERA NELLE 4 SEZIONI

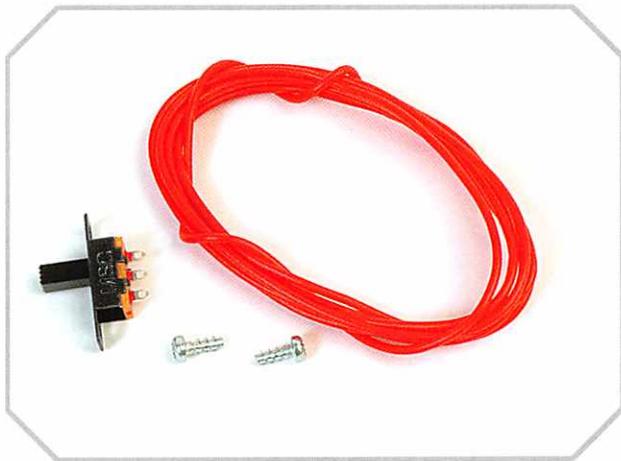
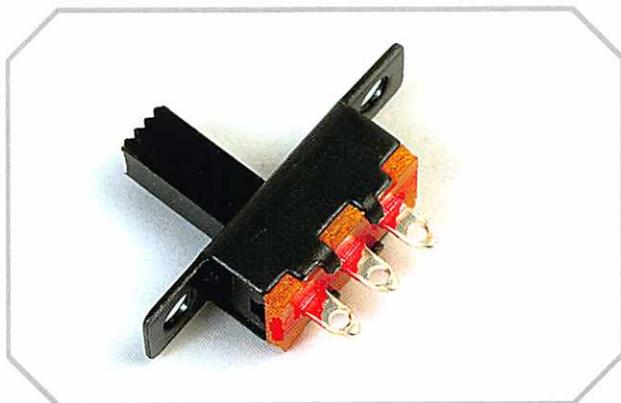
L'Opera è composta da 4 sezioni identificabili dalle fasce colorate, come indicato sotto. Le schede di ciascun fascicolo andranno suddivise nelle sezioni indicate e raccolte nell'apposito raccoglitore, che troverai presto in edicola. Per il momento, ti consigliamo di suddividere le sezioni in altrettante cartellette, in attesa di poterle collocare nel raccoglitore. A prima vista, alcuni numeri di pagina ti potranno sembrare ripetuti o sbagliati. Non è così: ciascuno fa parte di sezioni differenti e rispecchia l'ordine secondo cui raccogliere le schede. **Per eventuali domande di tipo tecnico scrivere al seguente indirizzo e-mail:** elettronicadigitale@microrobots.it

Hardware Montaggio e prove del laboratorio

Digitale di base Esercizi con i circuiti digitali

Digitale avanzato Esercizi con i circuiti sequenziali

Microcontroller Esercizi con i microcontroller

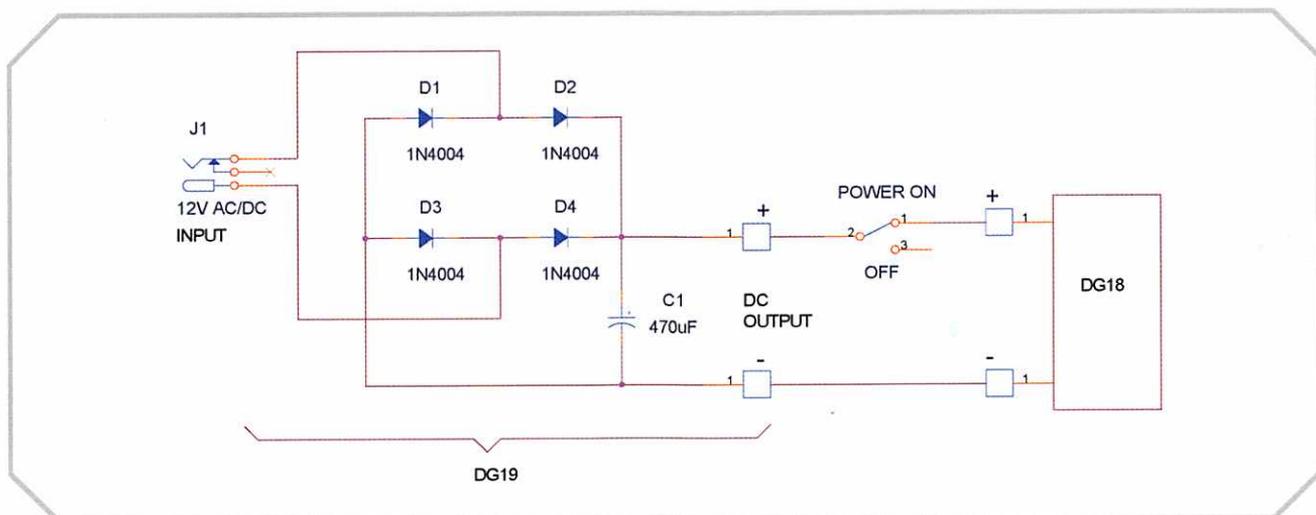
Commutatore POWER ON

Componenti allegati a questo fascicolo.

Dettaglio del commutatore.

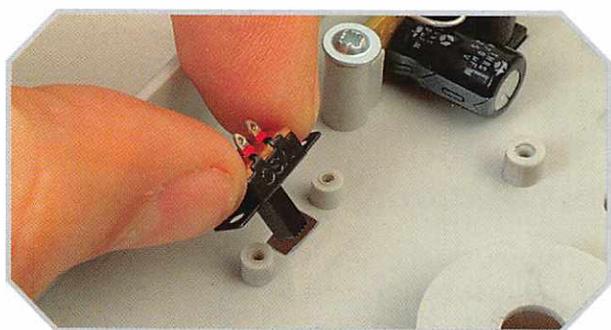
Con questo fascicolo viene fornito un commutatore a slitta, due viti necessarie per il montaggio e un pezzo di filo rosso per iniziare il cablaggio del sistema di alimentazione.

La funzione di questo commutatore è collegare o scollegare il positivo dell'alimentazione tra la scheda di ingresso DG19, che realizza la funzione di raddrizzamento e filtro, e il positivo della scheda DG18, che contiene i due stabilizzatori di tensione e il connettore dedicato al trasferimento dell'alimentazione tra i due pannelli principali del laboratorio.

Installazione

Il commutatore si installa dall'interno del pannello superiore, in modo che il suo comando fuoriesca dalla finestra rettangolare situata tra le etichette POWER ON e OFF, entrambe di colore rosso.

La finestra ha forma rettangolare, tuttavia in alcuni casi potrebbe risultare parzialmente chiusa da una piccola lamina di plastica derivante dalle operazioni di costruzione del pannello, è possibile rimuoverla con facilità utilizzando una lametta per tagliarla con precisione.

Schema elettrico dell'ingresso di alimentazione esterna.

In questa zona verrà montato il commutatore.

Montaggio delle viti del commutatore.

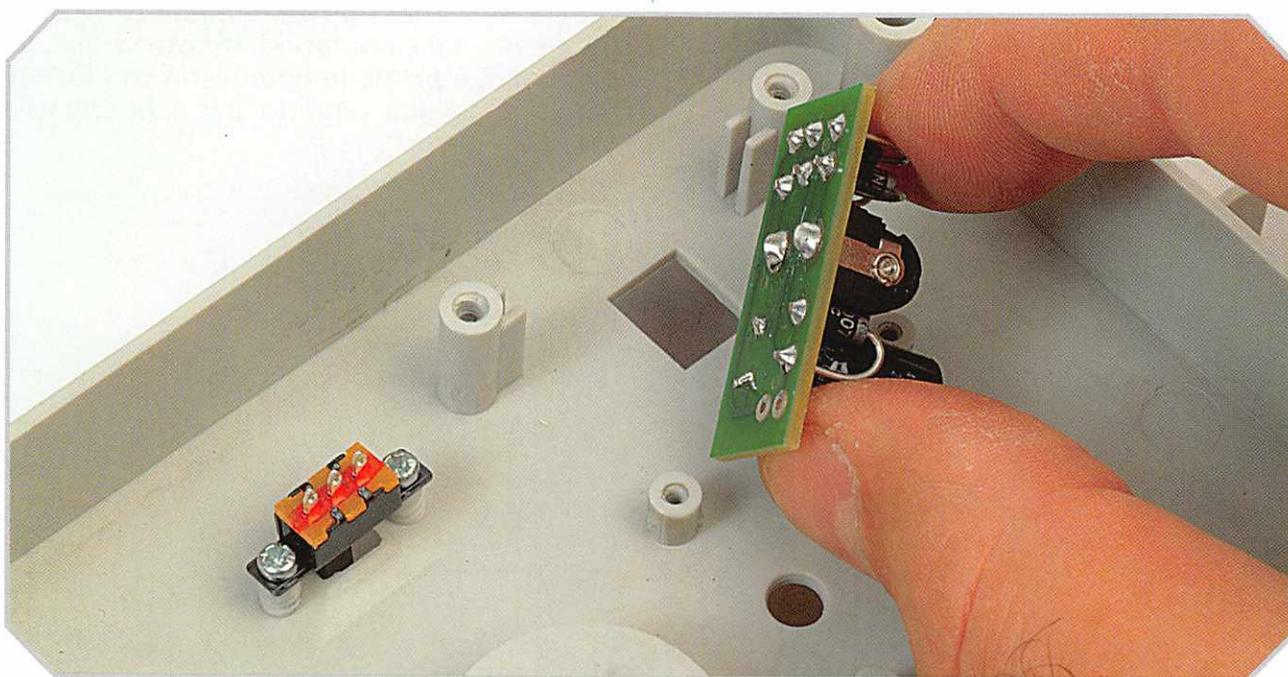
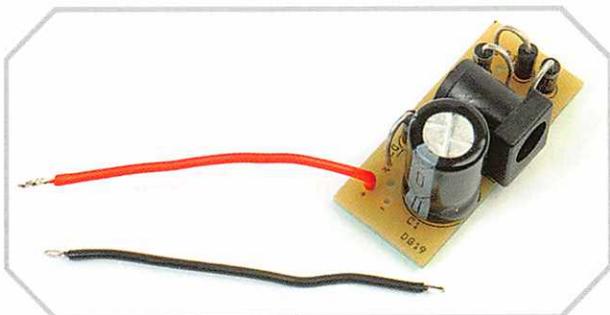
Il commutatore si fissa con le due viti che sono state fornite. All'inizio le avviteremo poco e dolcemente per non rovinare il filetto che la stessa vite crea sulla colonnina forata di plastica predisposta per l'alloggiamento della vite.

Copovolgeremo quindi il laboratorio per poter osservare il comando del commutatore dalla parte frontale del pannello superiore, il quale deve poter scivolare facilmente verso l'alto e verso il basso semplicemente azionandolo con un dito.

Collegamento della scheda DG19

Il commutatore appena installato riceve la tensione dal terminale positivo della scheda DG19, siglato come (+); il collegamento si realizza utilizzando il filo rosso fornito.

Questo collegamento si realizza con una saldatura, quindi toglieremo la scheda DG19 dal suo alloggiamento allentando le viti che la fissano, per non danneggiare nessuna parte del laboratorio.

Dopo aver fatto questo taglieremo un pezzo di filo nero della lunghezza compresa tra

Dopo aver fissato il commutatore si toglie DG19.

Scheda DG19 con un filo.

Scheda DG19 montata.

4,5 e 5 cm, e un pezzo di filo rosso della stessa lunghezza, asportando circa 4 mm di copertura plastica agli estremi di ogni filo per saldare il nero al terminale (-) e il rosso al terminale (+). In questo modo la scheda risulta cablata.

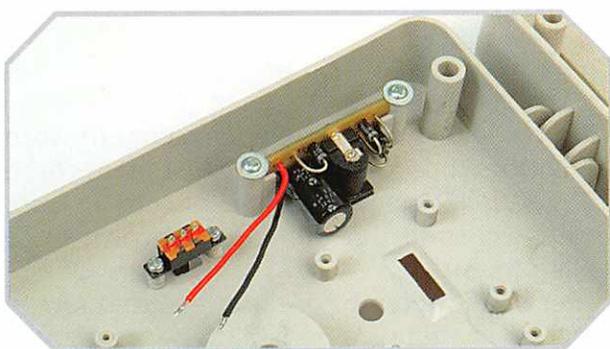
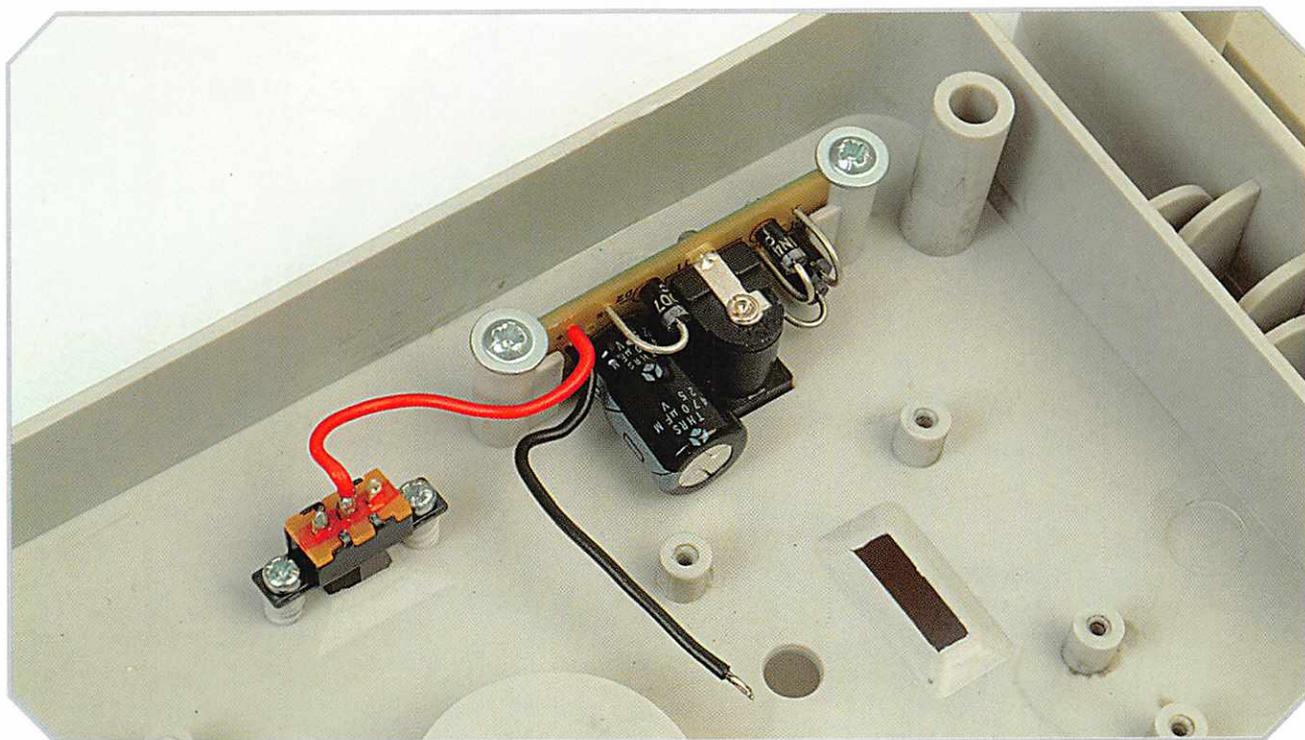
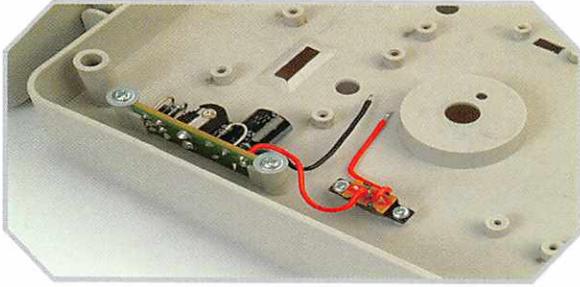
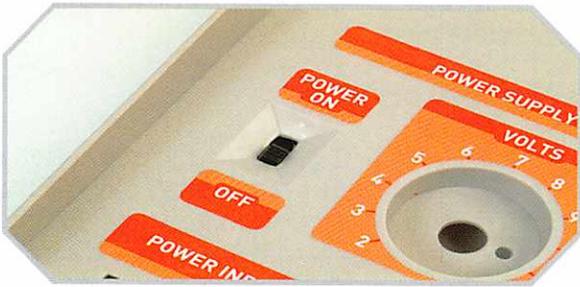
Montaggio di DG19

La scheda DG19 si può installare in modo definitivo fissandola con le sue due viti, che sono diverse dalle altre e presentano un allargamento sulla testa.

Collegamento

Taglieremo un altro pezzo di filo rosso della stessa lunghezza del precedente, ovvero tra 4,5 e 5 cm, e lo prepareremo nello stesso modo, togliendo un pezzo di copertura da ogni estremo per facilitare il collegamento tramite saldatura.

A questo punto continueremo a realizzare il collegamento, nel seguente modo:

– L'estremo del filo rosso che arriva dalla

Filo di ingresso della tensione al commutatore.

Il secondo filo rosso è quello di uscita.

Dettaglio del comando del commutatore.

scheda DG19 si salda sul terminale centrale del commutatore.

– L'estremo del secondo filo rosso preparato si salda sul terminale superiore del commutatore, il più vicino a POWER ON.

Queste saldature devono essere eseguite con attenzione, utilizzando stagno di qualità per applicazioni elettroniche e verificandole dopo averle realizzate.

Il laboratorio rimane, per il momento, con due cavi sciolti senza collegamento, in attesa di ricevere altro materiale con i prossimi fascicoli.

Precauzioni

Non bisogna collegare nessun alimentatore al connettore del pannello siglato come 12 V AC/DC, perché non è ancora in uso, inoltre, ci sono due fili sciolti ancora senza collegamenti.

Per ora dobbiamo continuare a utilizzare le batterie come unica possibilità per realizzare gli esperimenti.

Vista generale del laboratorio.

Pulsante ON/OFF

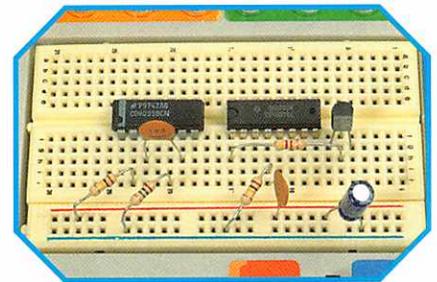
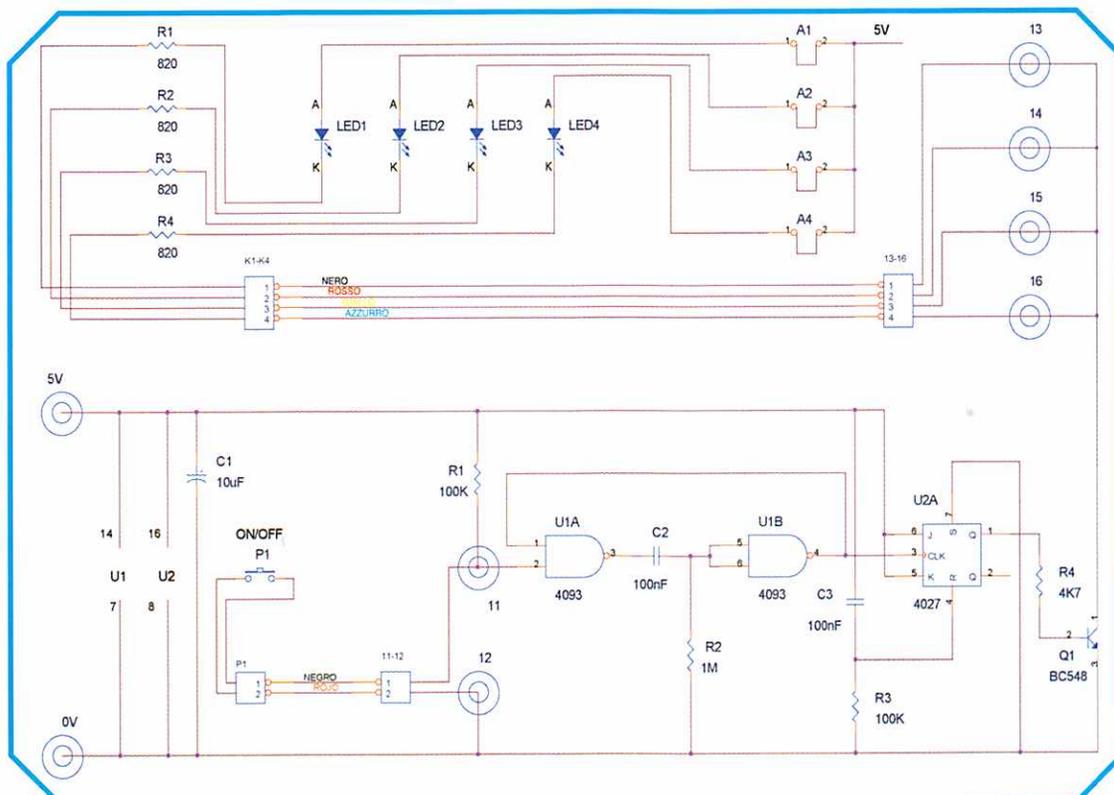
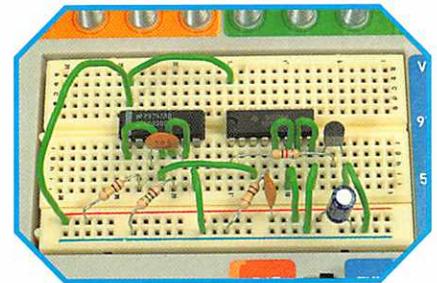
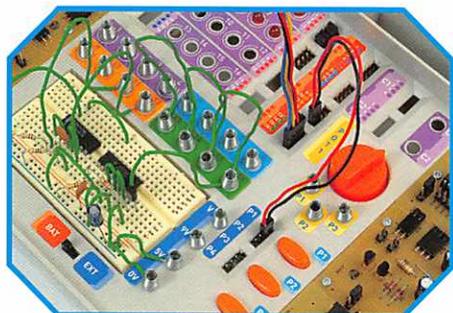
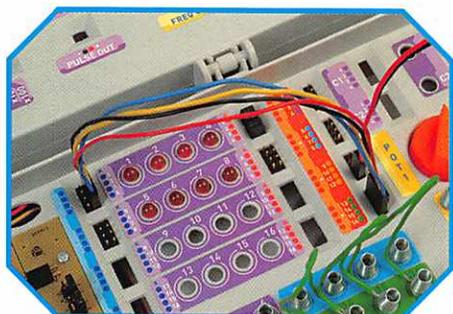
Questo circuito utilizza un flip-flop T il cui ingresso di clock serve per collegare e scollegare in modo sequenziale un circuito che, in questo caso, è rappresentato da 4 diodi LED.

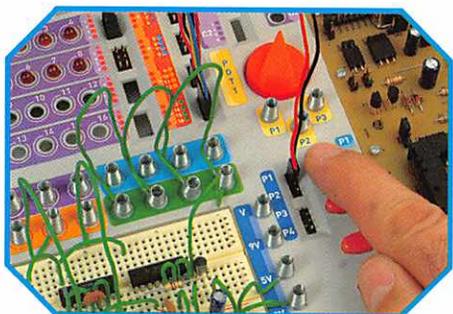
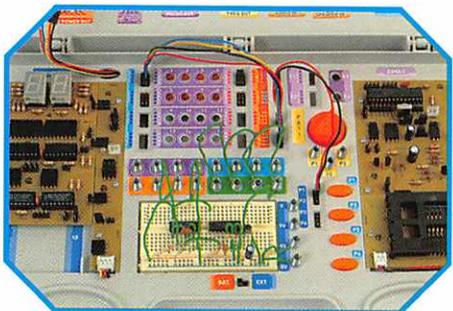
Idee

Per fare in modo che l'attivazione sia unica e il circuito non si spenga e si accenda diverse volte, è necessario assicurarsi che ogni attivazione del pulsante generi un unico impulso di clock, dato che l'uscita del flip-flop T, formata da uno dei flip-flop JK del circuito integrato 4027, U2A nello schema, cambia di stato ogni volta che riceve un impulso di clock.

L'impulso unico si ottiene eliminando i rimbalzi mediante un circuito monostabile, formato dalle porte U1A e U1B del circuito integrato 4093. Il tempo di durata del monostabile è determinato dai valori della resistenza R2 e del condensatore C2.

Questo montaggio risolve anche un altro problema di questo tipo di circuiti, la cui uscita può essere alta o bassa quando si collega-

Componenti sulla scheda Bread Board.

Cablaggio dei componenti della scheda.

Schema del circuito.

Il pulsante P1 si utilizza per la funzione ON e per quella OFF.

Il collegamento ai catodi dei LED si esegue con un cavetto a quattro fili.

Questi ponticelli collegano a 5 V gli anodi dei LED.

Il monostabile evita i rimbalzi generati premendo P1.

Esperimento completato.

no, cosa che può rappresentare un problema, ad esempio, quando viene ripristinata l'energia dopo una mancanza di tensione, perché non sappiamo in quale stato si troverà il circuito.

Questo problema si risolve con la rete RC, formata dal condensatore C3 e dalla resistenza R3. Vediamone il funzionamento, ricordando che un condensatore scaricato è praticamente un corto circuito virtuale al momento del collegamento, quindi, quando si collega l'alimentazione il terminale di RESET (R) si pone a livello alto il tempo sufficiente per fare in modo che l'uscita del flip-flop, terminale 1 del 4027, resti a livello basso e i LED non si illuminino dopo una mancanza di energia.

Per non caricare eccessivamente la porta e poter controllare una certa quantità di corrente, utilizzeremo un transistor la cui corrente di collettore può essere molto superiore a quella di uscita di una porta.

Montaggio

Il montaggio si esegue come d'abitudine, prestando particolare attenzione al collegamento del transistor Q1 e alla polarità del condensatore elettrolitico C1. Il collegamento del pulsante C1 si esegue con un cavetto a due fili e quello dei LED con uno a quattro fili. È molto importante collegare i quattro ponticelli tra gli anodi dei LED e il terminale del positivo posizionato sullo stesso connettore per facilitarne il collegamento.

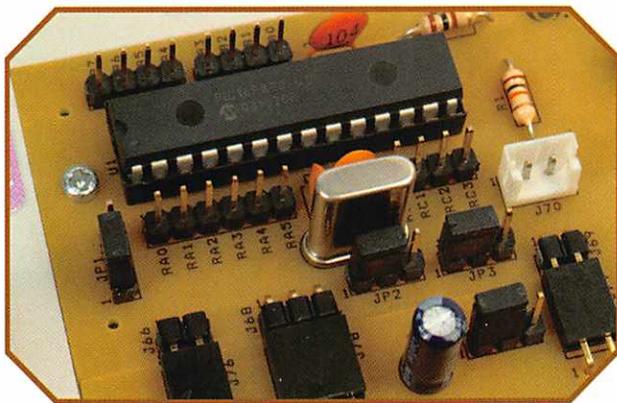
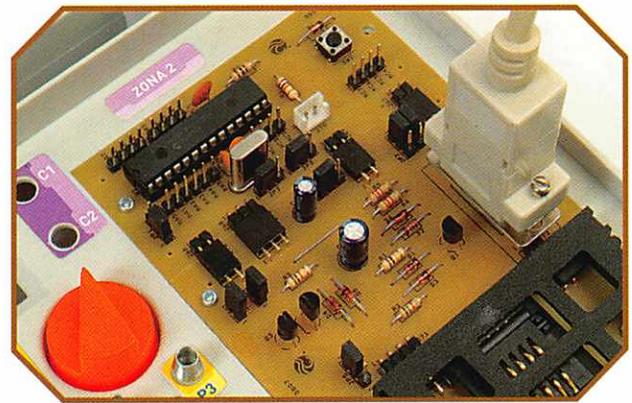
L'esperimento

Quando si collega l'alimentazione i quattro LED devono rimanere spenti. Premendo una volta P1 si devono illuminare, e se si preme nuovamente si spegneranno, in altre parole lo stesso pulsante si utilizza per l'accensione e lo spegnimento dei LED.

LISTA DEI COMPONENTI

U1	Circuito integrato 4093
U2	Circuito integrato 4027
R1, R3	Resistenza 100 K (marrone, nero, giallo)
R2	Resistenza 1 M (marrone, nero, verde)
R4	Resistenza 4K7 (giallo, viola, rosso)
C1	Condensatore 10 µF elettrolitico
C2,C3	Condensatore 100 nF

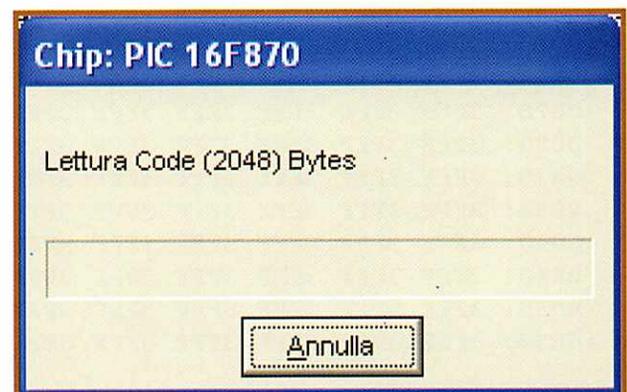
Esercizi 7, 8 e 9: la pratica

Negli esercizi di ripasso (microcontroller 91 e 92) abbiamo spiegato i programmi per lavorare con i temporizzatori TMR0 e TMR1, abbiamo progettato, compilato e anche simulato il codice, però ci manca ancora la cosa più importante: verificare il suo funzionamento nella pratica e che effettivamente esegua ciò che ci attendiamo. Dobbiamo quindi scrivere i programmi sul microcontroller ed eseguire i relativi montaggi.

Dobbiamo configurare i ponticelli della scheda per poter eseguire la scrittura.

Dopo aver collegato il Laboratorio al PC saremo pronti per scrivere.

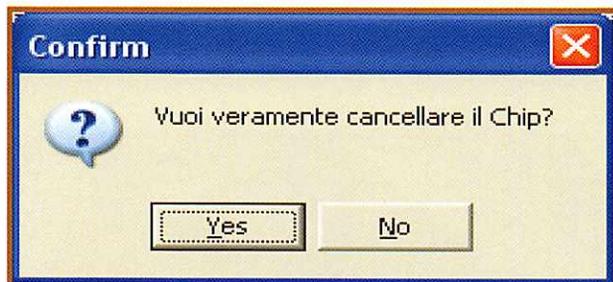
Configurazione hardware per IC-Prog

Per scrivere i programmi sul microcontroller utilizziamo il software IC-Prog, è necessario però che il laboratorio sia correttamente configurato in modo che il trasferimento si realizzi correttamente. Il microcontroller deve essere inserito sullo zoccolo nella posizione corretta e i connettori della scheda DG06: JP1, JP2 e JP3, devono avere i ponticelli nelle posizioni 1 e 2. Ricordate che per qualsiasi altra applicazione del microcontroller i ponticelli di questi connettori verranno posizionati sui terminali 2 e 3. La scheda avrà, in questo modo, la configurazione che possiamo vedere nella figura. I ponticelli JP8 e JP9 devono essere anch'essi collegati per scrivere il PIC e, in ultimo, dobbiamo anche collegare il cavo di scrittura tra il laboratorio e il PC.

IC-Prog

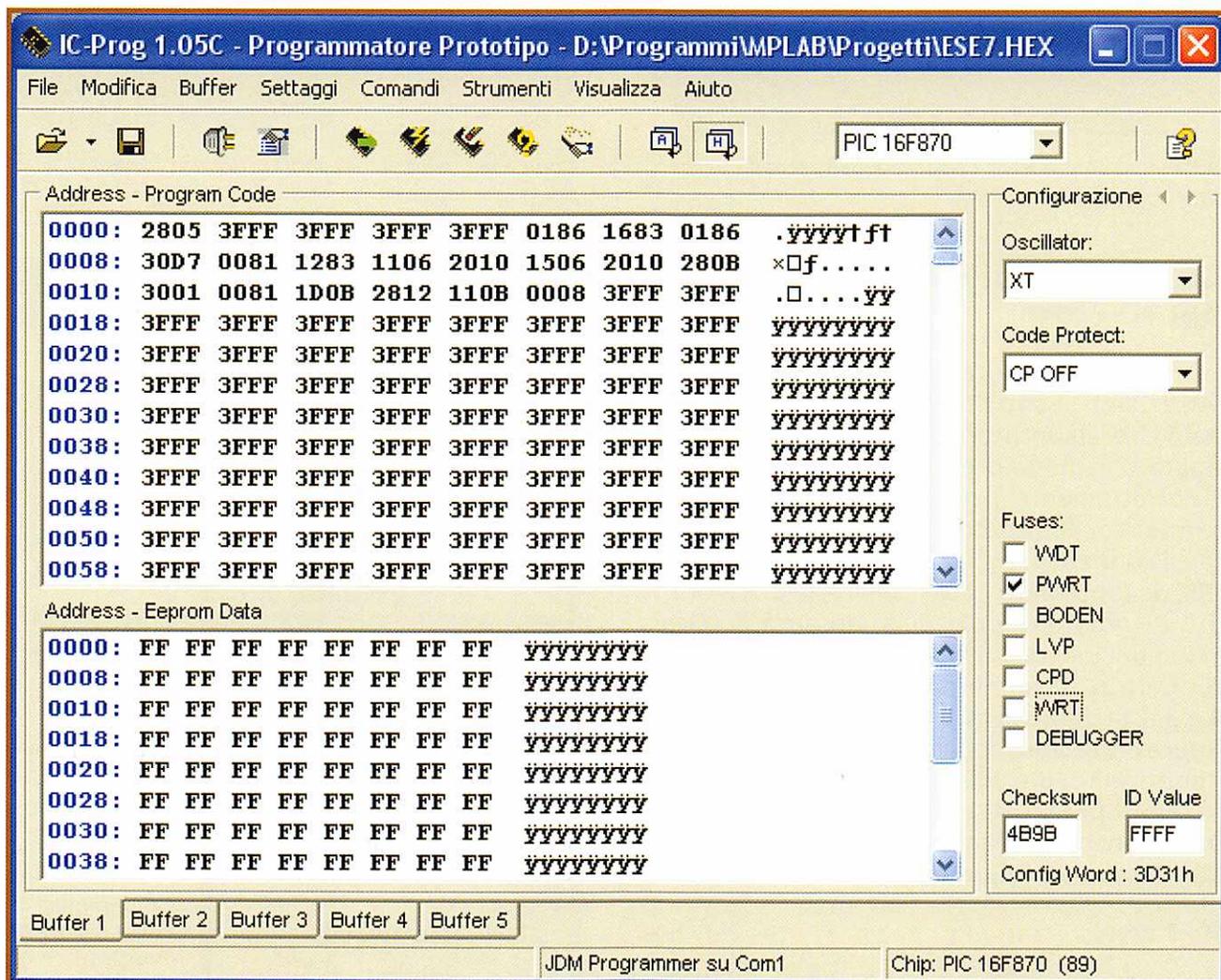
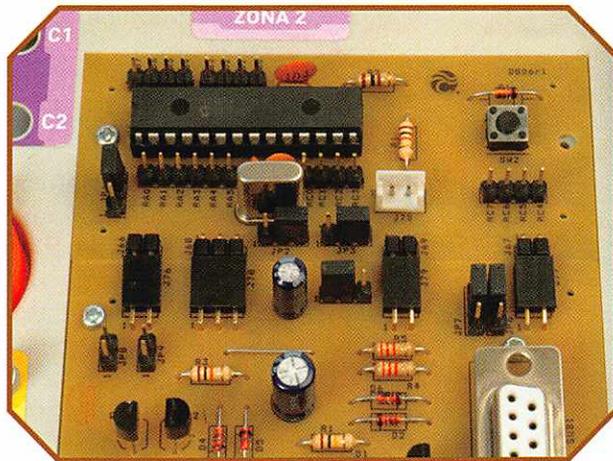
Inizieremo facendo partire IC-Prog e verificando di aver selezionato il PIC16F870 e che la

Finestra che indica lo stato del processo di lettura.

Il programma ci avvisa che il PIC è stato cancellato correttamente.

scrittura venga eseguita utilizzando la porta desiderata. Se osserviamo la parte inferiore della finestra del programma potremo vedere il programmatore e il dispositivo selezionato, nel nostro caso: "JDM Programmer su Com1"

Quando selezioniamo cancella e programma dobbiamo confermare la scelta.

e "Dispositivo: PIC 16F870 (89)". Se non compaiono questi valori dovremo configurare il programma adeguatamente.

La prima operazione che dovremo eseguire sul PIC sarà leggere il programma che esso contiene. A questo scopo selezioneremo l'opzione Leggi tutto dal menù Comandi, oppure premeremo F8 o selezioneremo l'icona corrispondente sulla barra degli strumenti. Se il PIC contiene un programma memorizzato cambieranno i valori della finestra "Program code". Il programma letto potrà essere salvato sul nostro PC se lo riterremo opportuno.

L'operazione successiva sarà cancellare il dispositivo, passaggio consigliabile prima di scrivere il programma. Selezioneremo a questo scopo Cancella tutto dal menù o con l'ico-

Aspetto di IC-Prog debitamente configurato e caricato con l'esercizio "ese7.hex".

Configurazione dei jumper per il funzionamento normale o di lavoro.

Montiamo il circuito comune ai due esercizi del TMR0.

na corrispondente sulla barra degli strumenti. Essendo questo un processo critico, apparirà una richiesta di conferma a cui risponderemo Yes, e in questo modo si cancellerà il dispositivo. Al termine di questo processo apparirà una finestra come quella riportata in precedenza che ci confermerà il successo dell'operazione. Per verificare il processo leggeremo nuovamente il dispositivo e controlleremo che in tutti gli indirizzi di memoria della finestra del codice di programma appaia il valore 3FFF. In alcuni casi è necessario configurare l'oscillatore e i bit della parola di configurazione, prima di procedere alla cancellazione, anche se questo non è molto comune. La corretta impostazione della parola di configurazione è invece realmente necessaria per realizzare la scrittura di un programma sul microcontroller.

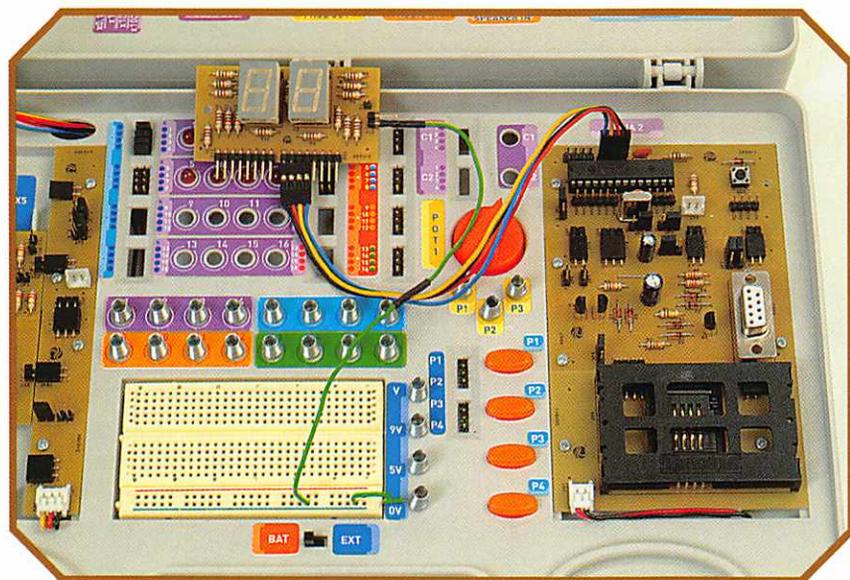
Scrittura

Dobbiamo selezionare il programma che vogliamo scrivere sul PIC. Apriamo il file desiderato (.hex) e verifichiamo come cambia la finestra che contiene il codice del programma. Prima di scrivere configuriamo l'oscillatore selezionando XT e la parola di configurazione selezionando i bit WDT e PWRT per l'esercizio "ese8.hex". Ricordate che dovremo mantenere l'opzione di protezione del codice su CP OFF. Selezionando Programma tutto caricheremo il programma selezionato sul PIC. Dato che anche questo è un processo critico ci verrà presentata una richiesta di conferma del processo a cui risponderemo Yes. Anche se otterremo un messaggio di verifica corretta, è consigliabile leggere nuovamente il dispositivo per verificare che il processo sia stato eseguito con successo.

Ora abbiamo a disposizione il microcontroller che contiene il programma desiderato. Per caricare qualsiasi altro programma dovremo ripetere i passi visti finora.

Montaggio dell'esercitazione con il TMR0

I due esercizi predisposti per lavorare con il TMR0 hanno lo stesso montaggio. La prima cosa da tener presente e che per lavorare con il Laboratorio in modo diverso dal trasferimento dei dati con il computer, dobbiamo cambiare la configurazione dei ponticelli. Ora JP1, JP2 e JP3 dovranno avere i ponticelli sui terminali 2 e 3, toglieremo i ponticelli dai JP8 e JP9 e scollegheremo il cavo di scrittura dal laboratorio e dal PC. Nelle immagini della figura possiamo osservare come deve essere configurato il laboratorio per lavorare con il microcontroller in modo funzionamento o di lavoro.

Uniremo mediante un filo il terminale RB2 con l'anodo di uno dei due diodi della matrice e collocheremo i ponticelli sui catodi dello stesso. Dopo aver fatto questo potremo fornire alimentazione e verificare se abbiamo risolto correttamente l'esercizio, controllando come lampeggia il diodo alla frequenza programmata.

Montaggio per l'esercitazione con il TMR1 "ese9".

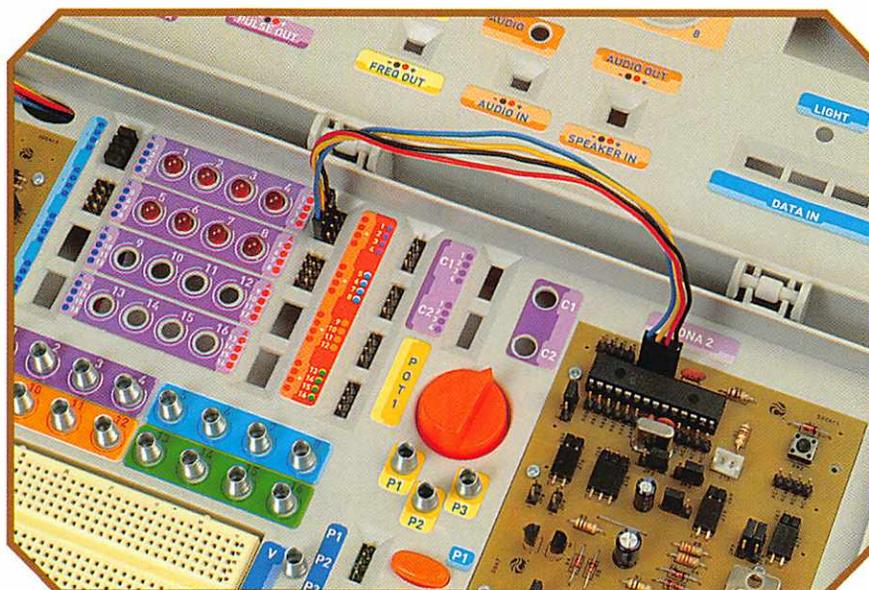
Esercizio 8

Se carichiamo sul PIC l'altro esercizio del TMR0, ovvero ese8, potremo verificare come anch'esso funzioni correttamente. Ricordate che per caricare il programma sul microcontroller è necessario configurare nuovamente l'hardware del laboratorio, aprire il software IC-Prog e caricare il programma con i passaggi che esso richiede. In questo caso il lampeggio sarà più lento e, la frequenza minore così come volevamo che fosse.

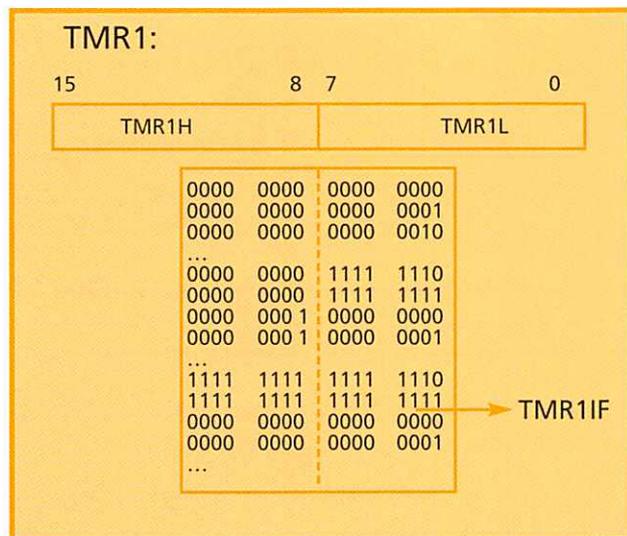
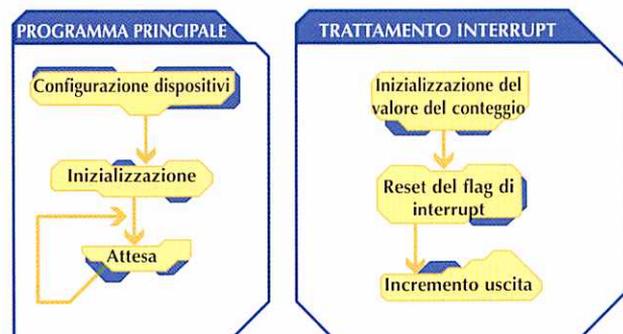
Montaggio dell'esercitazione con il TMR1

Per verificare la corretta soluzione di questa esercitazione dobbiamo eseguire il relativo montaggio, dopo aver correttamente caricato il programma "ese9" sul PIC. In questo caso utilizzeremo come uscita tutta la porta B, quindi dovremo utilizzare due cavetti di collegamento. Questa volta invece di collegarci alla matrice dei diodi ci collegheremo direttamente a uno di due display, J11 o J12. Nel caso in cui i cavetti non arrivassero al connettore, potremo staccare la scheda e avvicinarla facendo attenzione a non entrare in contatto con qualche elemento metallico che possa provocare un corto circuito. Dopo esserci assicurati che i ponticelli siano nella configurazione corretta per il modo funzionamento o modo lavoro (JP1, JP2 e JP3 con i ponticelli sui terminali 2 e 3, e JP8 e JP9 tolti), potremo fornire alimentazione e osservare il risultato.

Vediamo come i LED del display (o i LED della matrice) si accendono in modo sequenziale, rispondendo all'istruzione di incrementare di 1 il valore di uscita, che abbiamo utilizzato nel codice del programma.

Potete provare questo stesso esercizio collegando la porta B alla matrice dei diodi o provare a modificarlo utilizzando, ad esempio, le istruzioni di rotazione.

Nel laboratorio è più semplice lavorare con la matrice di LED che con il display.

Esercizio 9: il timer 1 (TMR1), il programma

In questo esercizio lavoreremo con il secondo temporizzatore, il TMR1. Tramite gli esercizi vogliamo farvi applicare le conoscenze acquisite e che impariate a maneggiare i potenti dispositivi del microcontroller. Sul secondo CD, il nome del file su cui si trova la soluzione di questo esercizio è "ese9.asm".

Il TMR1 è un temporizzatore/contatore da 16 bit che quando funziona come temporizzatore incrementa il suo valore ogni ciclo di istruzione (Fosc/4).

Organigramma delle applicazioni.

Introduzione

Il TMR1 può funzionare in tre modi diversi: temporizzatore, contatore sincrono e contatore asincrono. Dato che come contatore è necessario il collegamento di un oscillatore o di un clock esterno (secondo che sia sincrono o asincrono) proponiamo un esercizio in cui il TMR1 funziona come temporizzatore. In questo modo il suo funzionamento è molto simile a quello del TMR0, con la differenza che il range del predivisore in questo caso è: 1/1, 1/2, 1/4, 1/8.

Enunciato

Si vuole accendere sequenzialmente i LED di uno dei display a 7 segmenti collegato alla porta B ogni mezzo secondo.

Per realizzare questo esercizio abbiamo bisogno di tutte le linee della porta B per utilizzarle come uscite e anche di ripassare i nostri concetti riguardo al temporizzatore TMR1 e, nuovamente, quelli degli interrupt.

Gli organigrammi che rispondono a quanto richiesto nell'enunciato sono riportati nell'immagine a fianco. In uno si risolve il programma principale, in cui si configurano i dispositivi che vogliamo utilizzare: porta B, interrupt e temporizzatore, e successivamente entreremo in un ciclo senza fare nulla in attesa che si generi l'interrupt. Nella subroutine di trattamento degli interrupt caricheremo il valore del conteggio del temporizzatore, resettere-

```
                              ESERCIZIO: TMR1
                           -------------------
;
;
;Il programma muove in modo sequenziale i LED del 7 segmenti ogni mezzo secondo (500 ms).

;Si suppone che il TMR1 lavori con un prescaler di 1/8, a 4 MHz.
;Calcoliamo il valore che dobbiamo caricare sul TMR1:
;       500ms=4*(1/4MHz)*valore*8 -->   valore(dec)=62500
;       valore(bin)=11110100 00100100
;                   -------- --------
                      TMR1H    TMR1L
```

Nel confezionamento di un programma sono necessari i commenti.

```
LIST      P=16F870          ;Definiamo il nostro PIC
INCLUDE   "P16F870.INC"     ;File dei registri interni


ORG       0
GOTO      INIZIO
ORG       4
GOTO      INT
ORG       5
```

Inizieremo a sviluppare il codice con l'intestazione a cui siamo abituati.

```
;Programma principale.

INIZIO   clrf     PORTB
         bsf      STATUS,RP0       ;Passiamo al banco 1
         clrf     TRISB            ;Porta B uscite
         bsf      PIE1,TMR1IE      ;Abilitiamo l'interrupt del TMR1 (TMR1IE)
         bcf      STATUS,RP0       ;Torniamo al banco 0
         movlw    b'00110001'
         movwf    T1CON            ;Attiviamo il TMR1 e impostiamo il divisore da 1/8
         movlw    b'11000000'
         movwf    INTCON           ;Abilitiamo gli interrupt: globali e per il PEIE

NULLA    clrwdt                    ;Ciclo che non fa nulla
         goto     NULLA
```

Codice del programma principale.

```
;Programma di trattamento dell'interrupt

INT      movlw    b'11011100'      ;c2(00100100)
         movwf    TMR1L
         movlw    b'00001100'      ;c2(11110100)
         movwf    TMR1H            ;Carichiamo il TMR1 con il complemento di 31250 in hex.
         bcf      PIR1,TMR1IF      ;Impostiamo a zero il flag TMR1IF
         incf     PORTB,f          ;Incrementiamo il valore della porta B che riflette il
                                   ;risultato sul display a 7 segmenti
         retfie                    ;Ritorno da Interrupt
```

Codice della subroutine di trattamento dell'interrupt.

mo il flag di indicazione di interrupt e incrementeremo l'uscita per accendere così i LED del display.

Codice

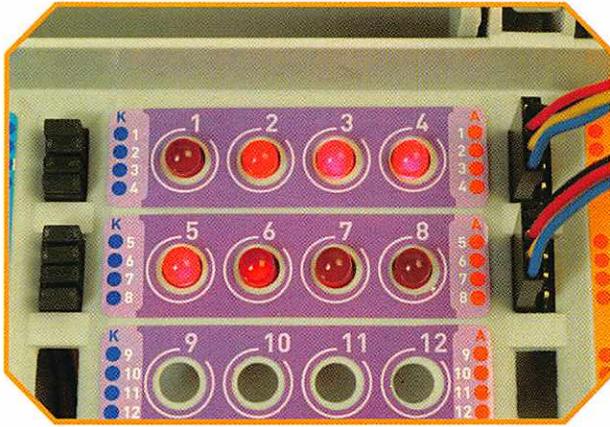
Il grado di complessità di questo esercizio non è molto elevato. L'unica difficoltà risiede nella gestione degli interrupt e del temporizzatore.

Inizio

Per iniziare con il programma scrivete i commenti che definiscono la funzionalità dello stesso. Vi consigliamo di scrivere tutto ciò che è necessario a far sì che in seguito risulti più facile risolvere il problema o interpretare il programma voi stessi o un futuro fruitore. Siamo pronti per iniziare il programma e, come sempre, inizieremo dalla definizione del microcontroller, dalla libreria che contiene la definizione dei dispositivi interni e dalle direttive del programma in cui definiremo gli indirizzi della memoria dove dovrà risiedere il nostro programma.

Programma principale

Secondo quanto previsto nell'organigramma, nel programma principale dovremo configurare i dispositivi. La porta B, quindi, si configura come uscita, abiliteremo gli interrupt del temporizzatore TMR1 impostando a 1 il bit TMR1IE del registro PIE1, abiliteremo questo temporizzatore indicando il predivisore di frequenza da utilizzare e abiliteremo gli interrupt globali necessari per qualsiasi interrupt si generi. Con questo passaggio abbiamo realizzato la configurazione di cui abbiamo bisogno per risolvere l'esercizio. L'ordine dei pas-

È molto facile imitare alcuni effetti che tempo fa risultavano sorprendenti, come il frontale di KIT (Supercar).

saggi non è determinante, ma dovrà essere sempre eseguito in modo da non richiedere inutili cambi di banco (in questo modo risparmieremo istruzioni), e dovrà avere una forma ordinata. All'inizio di questo blocco inseriremo un'istruzione di cancellazione della porta di uscita per assicurarci che quando caricheremo il programma sul microcontroller parta con tutti i LED spenti.

Infine entreremo in un ciclo infinito senza fare nulla attendendo che si generi un interrupt del temporizzatore.

Trattamento dell'interrupt

È in questa subroutine dove si risolve realmente il programma e dove il progettista può giocare con le diverse soluzioni o inserire le eventuali aggiunte. La prima cosa da fare è caricare il valore di conteggio del temporizzatore, il quale si ottiene in funzione del tempo che vogliamo controllare. Dobbiamo anche resettare o impostare a 0 il bit del flag di interrupt TMR1IF. In ultimo, definiremo ciò che vogliamo fare dopo che è trascorso il tempo, in questo caso incrementeremo il valore della porta B, in modo che questo valore si rifletta sui LED del display. A questo scopo potremo utilizzare diverse istruzioni e osservare i risultati relativi. Se vogliamo far accendere solamente un LED alla volta invece di accenderli uno dopo l'altro in sequenza, dovremo caricare un 1 e utilizzare le istruzioni di rotazione. Potremo utilizzare i LED della matrice invece di quelli del display, e simulare l'effetto di "Supercar", in modo che quando l'ultimo LED sarà acceso si realizzi la sequenza in senso inverso.

Non bisogna dimenticare di terminare il programma con la direttiva END. Nell'immagine in basso possiamo vedere il codice completo del programma.

Compilazione

Realizzato il programma, dobbiamo compilare il nostro codice per poter così correggere gli errori sintattici che potremmo aver commesso. È normale che, dopo aver progettato il codice, si trovi qualche errore durante la compilazione, a questo scopo MPLAB ci offre un potente compilatore.

Apriremo MPLAB, creeremo un nuovo progetto a cui aggiungeremo il nostro codice che sarà stato precedentemente salvato nella directory dei progetti di MPLAB. Se selezioneremo Project → Build All otterremo il risultato riportato nell'immagine della figura della pagina successiva. Il nostro codice non riporta nessun errore, questo significa che voi non dovrete far altro che correggere eventualmente qualche piccolo errore del programma che avrete realizzato. Il fatto di dover correggere qualche piccolo errore è positivo perché aiuta a mettere a punto e a chiarire concetti che potrebbero non essere stati bene assimilati.

Dopo aver ripulito il programma dagli errori e aver generato il file in codice macchina necessario per caricare il PIC, potremo simulare l'applicazione. La simulazione in un programma che lavora con interrupt non è molto semplice, ma vi consigliamo comunque di giocare

```
                        ESERCIZIO: TMR1
                      -------------------

;Il programma muove in modo sequenziale i LED del 7 segmenti ogni mezzo secondo (500 ms).

;Si suppone che il TMR1 lavori con un prescaler di 1/8, a 4 MHz.
;Calcoliamo il valore che dobbiamo caricare sul TMR1:
;       500ms=4*(1/4MHz)*valore*8 -->   valore(dec)=62500
;       valore(bin)=11110100 00100100
;                   -------- --------
;                    TMR1H    TMR1L

        LIST    P=16F870            ;Definiamo il nostro PIC
        INCLUDE "P16F870.INC"       ;File dei registri interni


        ORG     0
        GOTO    INIZIO
        ORG     4
        GOTO    INT
        ORG     5

;Programma di trattamento dell'interrupt.

INT     movlw   b'11011100'         ;c2(00100100)
        movwf   TMR1L
        movlw   b'00001100'         ;c2(11110100)
        movwf   TMR1H               ;Carichiamo il TMR1 con il complemento di 31250 in hex.
        bcf     PIR1,TMR1IF         ;Impostiamo a zero il flag TMR1IF
        incf    PORTB,F             ;Incrementiamo il valore della porta B che riflette il
                                    ;risultato sul display a 7 segmenti
        retfie                      ;Ritorno da Interrupt

;Programma principale.

INIZIO  clrf    PORTB
        bsf     STATUS,RP0          ;Passiamo al banco 1
        clrf    TRISB               ;Porta B uscite
        bsf     PIE1,TMR1IE         ;Abilitiamo l'interrupt del TMR1 (TMR1IE)
        bcf     STATUS,RP0          ;Torniamo al banco 0
        movlw   b'00110001'
        movwf   T1CON               ;Attiviamo il TMR1 e impostiamo il divisore da 1/8
        movlw   b'11000000'
        movwf   INTCON              ;Abilitiamo gli interrupt: globali e per il PEIE

NULLA   clrwdt                      ;Ciclo che non fa nulla
        goto    NULLA

        END
```

Codice del programma completo.

Risultato della compilazione.

MPLAB IDE - C:\PROGRA~1\MPLAB\PROGETTI\PRATICA3.PJT

File Project Edit Debug PICSTART Plus Options Tools Window Help

c:\progra~1\mplab\progetti\pratica3.asm

```
;                          ESERCIZIO: TMR1
;                          ------------------
;
;Il programma muove in modo sequenziale i LED del 7 segmenti ogni mezzo secondo (5

;Si suppone che il TMR1 lavori con un prescaler di 1/8, a 4 MHz.
;Calcoliamo il valore che dobbiamo caricare sul TMR1:
;        500ms=4*(1/4MHz)*valore*8 -->   valore(dec)=62500
;        Valore(bin)=11110100 00100100
;                    -------- --------
;                      TMR1H    TMR1L


         LIST     P=16F870          ;Definiamo il nostro PIC
         INCLUDE  "P16F870.INC"     ;File dei registri interni


         ORG      0
         GOTO     INIZIO
         ORG      4
         GOTO     INT
         ORG      5

;Programma di trattamento dell'interrupt

INT      movlw    b'11011100'       ;c2(00100100)
         movwf    TMR1L
         movlw    b'00001100'       ;c2(11110100)
         movwf    TMR1H             ;Carichiamo il TMR1
         bcf      PIR1,TMR1IF       ;Impostiamo a zero
```

Watch_1

Address	Symbol	Value
06	PORTB	B'00000000'

Stopwatch

Zero — Cycles: 0 — Time: 0.00 ns

Processor Frequency: 4.000000 MHz

Clear On Reset

Close — Help

Special Function Register Window

SFR Name	Hex	Dec	Binary	Char
w	00	0	00000000	.
tmr0	00	0	00000000	.
option_reg	FF	255	11111111	.
pcl	00	0	00000000	.
pclath	00	0	00000000	.
status	18	24	00011000	.
fsr	00	0	00000000	.
porta	00	0	00000000	.
trisa	3F	63	00111111	?
portb	00	0	00000000	.
trisb	FF	255	11111111	.
portc	00	0	00000000	.
trisc	FF	255	11111111	.
intcon	00	0	00000000	.
pir1	00	0	00000000	.
pie1	00	0	00000000	.
pir2	00	0	00000000	.
pie2	00	0	00000000	.
tmr1l	00	0	00000000	.
tmr1h	00	0	00000000	.
t1con	00	0	00000000	.
tmr2	00	0	00000000	.
pr2	FF	255	11111111	.
t2con	00	0	00000000	.
ccpr1l	00	0	00000000	.
ccpr1h	00	0	00000000	.
ccp1con	00	0	00000000	.
rcsta	00	0	00000000	.
txreg	00	0	00000000	.
rcreg	00	0	00000000	.
txsta	02	2	00000010	.
spbrg	00	0	00000000	.
adresh	00	0	00000000	.
adresl	00	0	00000000	.
adcon0	00	0	00000000	.
adcon1	00	0	00000000	.
pcon	00	0	00000000	.
eedata	00	0	00000000	.

35 Col 1 | 49 | WR No Wrap INS | PIC16F870 | pc:0x00 | w:0x00 | --z dc c | Bk On | Sim | 4 MHz | User

Videata di simulazione di MPLAB.

con il programma e di provare le diverse opzioni viste quando lo abbiamo studiato.

Conclusioni

Il programma è stato predisposto per essere scritto sul microcontroller.

Questo programma, i precedenti e alcuni che vedremo lungo il corso dell'opera possono essere utilizzati per fare pratica con altre istruzioni e nuovi dispositivi. Infatti in questo programma potrete utilizzare istruzioni di rotazione, lo potrete ampliare in modo che ripeta il ciclo per un numero determinato di volte, lo potrete combinare con altri (ad esempio accendendo i LED in modo sequenziale e quando si attiva un pulsante fornire un numero casuale), ecc. Per diventare un esperto programmatore è necessario dedicare un pò di tempo al software, avere il coraggio di uscire dal copione e improvvisare programmi nuovi.